Recovering the conductance of a resistor network

Victor Lee Michael Mills
University of Washington

Summer 1991

Abstract

This paper reports the findings in a series of experiments to determine the conductance of a physical network. The procedure is outlined by Curtis and Morrow in [1], [2]. In the experiments, we have tested Curtis and Morrow's reconstruction schemes for the network as well as the Dirichlet to Neumann map (Λ_Γ map). We also verify the uniqueness of the Dirichlet solution. We find various properties of the Λ_Γ matrix, and new techniques to reconstruct the network as well as the Λ_Γ map.

I. Introduction

A. Physical Problem

The physical problem of interest is to compute the conductance of a two dimensional object. Physically, there isn't a definite way to measure the conductance of a two or higher dimensional object. In order to measure the conductance of a piece of material, one would use approximation techniques: First, subdivide the object into small square regions and approximate each region by a resistor network. Second, determine the conductance of the resistor network. Third, use the resulting conductance to approximate that of the square region. Finally, obtain an approximation to the conductance of the two dimensional object by averaging all square regions. However, there are some subtle problems to be addressed. For example, how can one combine the resulting conductances of a discrete network to formulate the conductance for a continuous region. In this paper we are not going to deal with these subtle problems. We report all the findings in trying to reconstruct the physical network. Our reconstruction scheme is based on Curtis and Morrow's alogrithm in [1]. The remaining sections of this introduction outline how the reconstruction alogrithm was developed.

B. Continuous Problem

In [1], Curtis and Morrow pointed out that the alogrithm to find the conductance of a physical network originated from the continuous inverse problem related to γ-harmonic functions. The continuous inverse problem is: Given the Dirichlet to Neumann map of a γ-harmonic function, u, on a region Ω, find the scalar function γ.

The γ-harmonic function u is the potential function on Ω, where Ω is the region on which the conductance is to be determined. γ is a positive scalar function on Ω; it represents the conductance of the region. A γ-harmonic fuction, u, in Ω is the fuction that satisfies $\gamma\nabla^2 u = 0$ on Ω. Curtis and Morrow defined the Dirichlet to Neumann map, Λ_Γ, as the map that takes the data from the Dirichlet problem to that of the Neumann problem. The Dirichlet problem of γ-harmonic function is: find a γ-harmonic function u on a domain Ω, if its values at the boundary of Ω are given. The Neumann problem provides the normal derivative of a γ- harmonic function u, and asks to find u on Ω.

One special feature of the Dirichlet and the Neumann problems is the uniqueness of their solutions. To prove the uniqueness of the Dirichlet problem, one first needs to show that if u is a γ-harmonic function on Ω and $u = 0$ on $\partial\Omega$, and γ is a positive scalar function, then $u \equiv 0$ on Ω. Then, one will use a proof by contradiction to show that if u and v are distinct solutions to the same Dirichlet problem, then $u - v = 0$ on Ω and by the fact proven at the beginning, one can conclude that $u \equiv v$ over Ω and that is a contradiction.

With these definitions, we can now move on to defining the forward and inverse problem associated with the continuous case. The forward problem is: Given a γ-harmonic function on a domain Ω in R^{n}, and a positive scalar function γ, find the Dirichlet to Neumann map. The inverse problem is: Given the Dirichlet to Neumann map and the domain Ω in R^{n}, find the scalar function γ.

C. Discretization of the Continuous problem

Curtis and Morrow's alogrithm to recover the conductance of resistors in a resistor network is based on discretizing the forward and inverse continuous problem stated above. [1] To discretize the continuous problem, one needs to define the equivalance of $\gamma\nabla^2 u = 0$ in the discrete case. Curtis and Morrow suggested the use of finite difference approximation method. Suppose the following is a physical network which approximates the rectangular region in the continuous case.

Figure 1. Schematic diagram for discretizing the continuous region.

Applying Finite difference method, we can approximate $\gamma\nabla^2 u = 0$ by $(\gamma u_x)_x + (\gamma u_y)_y = 0$. Inspecting figure 1 in the previous page, we realize that γu_x, and γu_y can be rewritten as:

$$\gamma u_x|_A = \gamma(A)[u(E) - u(p)]/h$$

$$\gamma u_x|_C = \gamma(C)[u(p) - u(W)]/h$$
$$\gamma u_y|_B = \gamma(B)[u(N) - u(p)]/h$$
$$\gamma u_y|_D = \gamma(D)[u(p) - u(S)]/h$$

for points A, B, C, D. where u(p) is the potential at point p, and h is the length between p and any neighboring node.

Applying the definition of derivatives in vector calculus again, we obtain the following expressions:

$$\gamma(u_x)_x = \frac{1}{h^2}(\gamma u_x|_A - \gamma u_x|_C)$$

$$\gamma(u_y)_y = \frac{1}{h^2}(\gamma u_y|_B - \gamma u_y|_D)$$

After substituting the expression for γu_x at point A, B, C, and D, we get the following expressions:

$$(\gamma u_x)_x = \frac{1}{h^2}[\gamma_A(u_E - u_p) + \gamma_C(u_W - u_p)]$$

$$(\gamma u_y)_y = \frac{1}{h^2}[\gamma_B(u_N - u_p) + \gamma_D(u_S - u_p)]$$

Now, we can write the condition for γ-harmonicity $(Div(\gamma\nabla u) = 0)$ in the discrete case as

$$\gamma_A(u_E - u_p) + \gamma_B(u_W - u_p) + \gamma_C(u_N - u_p) + \gamma_D(u_S - u_p)] = 0$$

After staring at it for a while, we can put the discrete condition in a more condensed form.

$$\left[\sum_{q \in itN(p)} \gamma(pq)\right] u(p) = \sum_{q \in N(p)} \gamma(pq) u(q)$$

Moreover, it is very interesing to note that the previous expression is exactly Kirchhoff's current law at a node when it is rewritten in a sightly different form:

$$\sum_{q \in N(p)} \gamma_q(u_q - u_p) = 0 \Rightarrow \sum Current\ enter\ a\ node = 0$$

where N(p) are the neighboring nodes of p.

Having discretized the condition for γ-harmonicity, we can now define the discrete forward and inverse problem. The discrete forward problem is: Suppose a potential function u satisfies

$$\sum_{q \in N(p)} \gamma_q(u_q - u_p) = 0$$

where $N(p)$ are the neighboring nodes of p, for all nodes in the physical network. Given an $n \times n$ physical network with know conductance γ, find the Λ matrix - the discrete approximation of the Λ_Γ map. The discrete inverse problem is: Given the Λ matrix of a physical network, find the conductance of the resistors inside the network.

D. Construction of the Physical Network

To test the validity and practicality of applying of the reconstruction algorithm of Curtis and Morrow for an $n \times n$ resistor network, we constructed a working model of such a network. We will refer to the Dirichlet to Neumann map for Ω_p as Λ_Γ.

Our network, henceforth referred to as Ω_p, was a 4×4 network, so its construction required 40 resistors (there are $2n(n+1)$ resistors in an $n \times n$ network). We chose 100 ohm(Ω) resistors, each rated at 5% tolerance. Thus we expected to have approximately uniform conductances throughout our network. Of course, the resistors did not provide uniform resistance, and, in fact, were not all within their rated tolerance. A diagram of Ω_p appears in Figure 2, with resistances (recall, $\gamma = \frac{1}{R}$) printed where they occur. Note also that the boldface numbers that appear on the boundary of the network are simply the numbers assigned by us for easy labeling.

```
                4                 3                 2                 1
                |                 |                 |                 |
              102.7             104.6             102.5             102.0
                |                 |                 |                 |
5 – 106.8   – – + – –   101.3   – – + – –   104.8   – – + – –   105.8   – – + – –   102.1 – 16
                |                 |                 |                 |
              106.7             102.5             105.7             101.9
                |                 |                 |                 |
6 – 104.1   – – + – –   104.7   – – + – –   105.5   – – + – –   104.5   – – + – –   104.6 – 15
                |                 |                 |                 |
              107.8             107.9             102.2             103.7
                |                 |                 |                 |
7 – 106.7   – – + – –   103.3   – – + – –   106.1   – – + – –   102.6   – – + – –   101.8 – 14
                |                 |                 |                 |
              102.1             104.9             105.3             106.3
                |                 |                 |                 |
8 – 101.5   – – + – –   103.3   – – + – –   109.4   – – + – –   103.7   – – + – –   102.0 – 13
                |                 |                 |                 |
              106.2             102.4             102.9             104.3
                |                 |                 |                 |
                9                 10                11                12
```

Interior values are in *ohms*.
Boundary values are labels for the nodes.

Figure 2. The physical network, Ω_p.

The network was built on a copper-clad circuit board, upon which 60 nodes were etched. Between each node a pad of copper was left to facilitate the later addition of capacitors (or diodes, transistors, etc.), either in series or parallel with the resistors. Four holes were drilled in each node, and the resistors were soldered into them to insure connections with low resistance.

The power supply consisted of six 1.5 V flashlight batteries connnected in series, providing a total potential differenceof about 9 V. The negative pole of the battery was connected directly to the ground of the network, while the positive side was connected to a 22 kΩ rheostat. This allowed us to vary the voltage across Ω_p from ≈ 9.02 V to $\approx 130\mu$V .

The ground lead was connected to 15 alligator clips so that all but one of the boundary nodes of Ω_p could be grounded at the same time ($u_q = 0$, where u_q is the potential at node $q \in \partial\Omega_p$). The positive lead was attached to a single alligator clip, so that it could be moved as different columns in Λ_γ were measured.

Metering was done using an Eico ammeter, Circuitmate voltmeter, and at times a Fluke 123 meter. The Eico was used primarily for measuring boundary currents, and provided measurements to the nearest 10^{-5} ampere. However, it was also found to have a systematic error on the order of -7%. The Circuitmate meter gave readings to the nearest 10^{-6} volt, with assumed high accuracy (we had no way of testing its correctness). The Fluke's range was similar to that of the Eico instrument, but its systematic error appeared greater, and few measurements were made with it.

However, systematic error is relatively unimportant in this application. If all of our readings were off by a common factor, it would be elementary to repair them. In our experiment, though, the numbers were inconsistent with *each other*, so no quick repair algorithm was applicable. More details about reconstructing the Λ matrix are given in Section IV.A.

II. Forward problem

In this section, we discuss the forward problem associated to a physical resistor network. First, we report the results of testing the uniqueness of the Dirichlet and the Neumann solution in the discrete case. Second, we show our finding about the insensitity of the lambda matrix.

A. Uniqueness of the Dirichlet solution

In the introduction, we have shown that the solution to the Dirichlet and the Neumann problem is unique mathmatically. In this section, we discuss the result of the experiments conducted to verify the uniqueness of the Dirichlet problem.

The experimental procedure involves three steps:

1. We connect the 4×4 model network constructed according to section I.D to an arbitary voltage input, and then measure the potential at each interior node.

2. We use the forward solver - a computer program that solves the Dirichlet and Neumann problem - to compute the theoretical potential of the network.

3. We compare the two sets of data to determine whether the physical measurements agree with the predicted data.

The data are reproduced at Appendix A. From these experimental data, we find the measured values agree with the predicted values up to the last digit of the meter's measurement. Thus, we believe that the solution to the Dirichlet problem is unique.

We were not able to test the uniqueness of the Neumann solution, since that kind of experiment involves using a constant current source, and this kind of input source is not availible to us.

B. Insensitivity of Lambda Map

We performed an experiment to show how independent the Λ-matrix is from the function of conductances, γ, on a network Ω. We created a hypothetical resistor network, Ω_h, with certain beginning conductances, and sent it to our forward solver to obtain the Λ matrix, Λ_Γ. We then altered some of the the values of γ, and recomputed Λ_Γ. The differences in the two results were barely noticeable, until a great deal of the network was changed. The graphical results of alterations made to a 15×15 network are given in Appendix C.

III. Physical Measurements

Since one of our goals was to determine the values of the conductances, γ, of the physical network, we had to create a Λ matrix from physical measurements around the boundary of the network. Taking all the measurements for the Λ matrix associated with our 4×4 network, Ω_p, would require $(4 \cdot 4)^2$ $(= 256)$ separate measurements. Each measurement involves moving the power and meter leads. We found this to be prohibitively time consuming, as doing the measurements once took almost three hours. Also, as will be shown in section IV of this paper, these measurements lacked the precision necessary to form a real Λ matrix. Tables of our measurements, as well as the computed Λ_Γ are listed in appendix B.

To save time, we implemented a method to reconstruct the Λ matrix from a reduced parameter set, as described in [2]. This parametrization allowed us to make fewer measurements — a total of $2n(n + 1)$, (one measurement for each $\gamma \in \Omega_p$). Hence, for our network, we only needed to make 40 measurements to fill a parameter set. The entries we chose were as follows:

$$\begin{array}{|cccc|cccc|cccc|cccc|}
\hline
\circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ \\
\circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ \\
\circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ \\
\circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ \\
\hline
\bullet & \bullet & \bullet & \bullet & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ \\
\circ & \bullet & \bullet & \bullet & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ \\
\circ & \circ & \bullet & \bullet & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ \\
\circ & \circ & \circ & \bullet & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ \\
\hline
\bullet & \bullet & \bullet & \bullet & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ \\
\bullet & \bullet & \bullet & \bullet & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ \\
\bullet & \bullet & \bullet & \bullet & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ \\
\bullet & \bullet & \bullet & \bullet & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ \\
\hline
\circ & \circ & \circ & \bullet & \circ & \circ & \circ & \bullet & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ \\
\circ & \circ & \bullet & \bullet & \circ & \circ & \bullet & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ \\
\circ & \bullet & \bullet & \bullet & \circ & \bullet & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ \\
\bullet & \bullet & \bullet & \bullet & \bullet & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ & \circ \\
\hline
\end{array}$$

●= parameter location.
○ = empty location; to be computed from parameters.

Figure 3. Parameter entries in partial Λ matrix.

To increase the precision of our measurements, we took a small statistical sampling of each boundary current measurement. That is, we measured the current at each node for at least ten different applied voltages. We then used linear regression on the ten values to interpolate a value for one volt. For example: To find the (5,1) entry in the Λ matrix, we grounded nodes 2 through 16, (set $u_q = 0$ for $q \in \partial\Omega_p$), and connected the positive lead at node 1 (set $u = \kappa$). Then, by adjusting the rheostat on the power supply, we varied κ between about 9 volts and .1 volt. The current (I) was then measured at node 5 for ten different values of κ, and the ten values were sent to a linear regression routine to find I for $\kappa = 1.0$.

However, even with the extra digits of precision that this method offered, our numbers were not consistent enough to provide a parameter set for Λ_Γ. In section IV.A, we explain why this is the case, and what level of accuracy is necessary to create a Λ matrix from a given set of parameters.

IV. Inverse Problem

A.1 Reconstruction of Λ Matrix from Parameters

It is proven in detail in [2] that a complete Λ matrix for an $n \times n$ network can be constructed from $2n(n+1)$ carefully chosen parameters. This is the method we used to reconstruct Λ_Γ. In summary, by solving sytems of equations formed from certain entries of the parameter set, one can take linear combinations of corresponding rows to fill in missing entries in the matrix.

	N	W	S	E
N	G	A	B	C
W	A^T	H	E	D
S	B^T	E^T	I	F
E	C^T	D^T	F^T	J

Figure 4. Block structure of $\overline{\Lambda}$. This diagram is reproduced from [2], to preserve notation.

We will refer to different blocks of the Λ matrix for our Ω_p network by the following scheme: Break the matrix up into 16 sections, by taking groups of n rows and n columns. By the definition of the Λ matrix in [2], each one of these $n \times n$ blocks contains the value of the current flowing out of the nodes on one side of the network (N,S,E,W) when $u = 1.0$ at the nodes on another face. For example, the values in block A are the currents at the boundary nodes along the north side of Ω_p ($A_{i,j} = I_j$, $i = (n+1)..2n$, $j = 1..n$), when

$u_k = 1$ for $k \in [n+1, 2n]$ and $u_k = 0$ for $k \notin [n+1, 2n]$. Figure 4 illustrates this block structure. Note that each block below the main diagonal is the transpose of the corresponding block above it (A and A^T, for example), and the blocks on the main diagonal (G, H, I, J) are symmetric $n \times n$ matrices. Now, we call this "blocked" lambda matrix $\overline{\Lambda}$.

Since the blocks in $\overline{\Lambda}$ represent portions of Λ_Γ, we must clarify the indexing of the two matrices. First, it is clear that $\overline{\Lambda}_{i,j} = \Lambda_{\Gamma i,j}$. However, within $\overline{\Lambda}$, the blocks are indexed differently, just as if they were smaller matrices superimposed over the larger one. Hence, for example, $\overline{\Lambda}_{1,1} = G_{1,1}$, and $\overline{\Lambda}_{4,12} = B_{4,4}$.

8	4	4	3	●	2	2	2	●	●	●	●	1	1	1	●
4	8	7	3	●	●	2	2	●	●	●	●	1	1	●	●
4	7	8	3	●	●	●	2	●	●	●	●	1	●	●	●
3	3	3	3	●	●	●	●	●	●	●	●	●	●	●	●
●	●	●	●	8	3	3	3	3	3	3	3	3	3	3	●
2	●	●	●	3	8	5	5	5	5	5	5	5	5	●	4
2	2	●	●	3	5	8	5	5	5	5	5	5	●	5	4
2	2	2	●	3	5	5	8	5	5	5	5	●	5	5	4
●	●	●	●	3	5	5	5	8	5	5	5	5	5	5	4
●	●	●	●	3	5	5	5	5	8	5	5	5	5	5	4
●	●	●	●	3	5	5	5	5	5	8	5	5	5	5	4
●	●	●	●	3	5	5	5	5	5	5	8	5	5	5	4
1	1	1	●	3	5	5	●	5	5	5	5	8	5	5	4
1	1	●	●	3	5	●	5	5	5	5	5	5	8	5	4
1	●	●	●	3	●	5	5	5	5	5	5	5	5	8	4
●	●	●	●	●	4	4	4	4	4	4	4	4	4	4	8

● = members of parameter set.
Z = the number of the stage at which the corresponding entry is computed.

Figure 5. The order of reconstruction of the Λ matrix from parameters.

Our program to reconstruct Λ_Γ from the parameter set illustrated in figure 3 uses the following order, also illustrated in figure 5:

1. Calculate all values above the main antidiagonal in block C, and copy their values to C^T.

2. Calculate values above the main diagonal in A; copy these to A^T.

3. Fill in the values for row $n+1$ above the main diagonal of $\overline{\Lambda}$, as well as row n below that diagonal. Copy these values to their corresponding locations across the diagonal.

4. Fill in the values for column $4n$ above the main diagonal, as well as column 1. Copy these values symmetrically also.

5. Finish filling in blocks D, E, H, F, and J above the main diagonal. Copy their values to their respective transpose matrices.

6. Fill in block I above the main diagonal and copy the values to the lower part.

7. Fill in the last entries of G, for $i \neq j$, and $i, j \notin [1, n]$.

8. Finally, for each row, i, the diagonal entry is found by summation:

$$\overline{\Lambda}_{i,i} = -\sum_{\substack{j=1 \\ j \neq i}}^{4n} \overline{\Lambda}_{i,j} \tag{1}$$

A.2 Requirements for Parameters

It is shown by Curtis and Morrow in [2], that there are several determinental conditions that must be met by the parameter set to insure that they will yield a valid Λ matrix upon reconstruction.

Specifically, the determinants of the small matrices used to calculate the non-parameters must have the **Right Sign**, which is defined as follows:

Definition 1: A determinant of a $n \times n$ matrix A in Λ_Γ is said to have the **Right Sign (RS)** if and only if:

1. $\det(A) < 0$ for $n \equiv 1, 2 \bmod 4$.

2. $\det(A) > 0$ for $n \equiv 0, 3 \bmod 4$.

The measured parameters from Ω_p yielded some determinants that did not have the **RS** property. Namely, the 3×3 matrix with upper left component at $\Lambda_{\Gamma 2,12}$ (call this small matrix M_1) had a negative determinant, when, by the **Right Sign** condition, it should have been positive. Likewise, the partially computed 3×3 matrix at $\Lambda_{\Gamma 6,2}$ (call this one M_2) had negative determinant. These two matrices could be adjusted, without prior knowledge of the actual values desired, so that their determinants had the **Right Sign** property. When this was done, the Λ matrix was much closer to a viable Λ_Γ, computed from the forward solver. However, it was not good enough to allow a reconstruction of Ω_p.

A.3 Repair Methods for Ω_p

As mentioned before, Curtis and Morrow give a procedure in [2] to construct a set of parameters for the Λ matrix, which we have followed. They give an order for filling in the actual parameter set by which one checks the determinants of several matrices as the parameters are added to the set. We did not use this check system, so our parameters are not quite consistent with each other—hence, our Λ_Γ is invalid.

However, the method of Curtis and Morrow gave us a procedure for repairing bad determinants in our reconstructed matrix. As the parameters are placed in block A, it is filled in starting from the lower left corner, and proceeding diagonally up and right, so that $A_{1,4}$ is filled in last. Hence, the last parameter to be checked in M_1 is the component in the upper right corner. So by sdjusting its value, $\det(M_1)$ can easily be given the **Right Sign**. We applied this technique to our parameters and found an improvement in our reconstructed Λ matrix. Likewise, the last parameter to be placed in M_2 is the upper right entry. By adjusting this number, we should be able to correct $\det(M_2)$. However, we found that it was necessary a to change a different parameter than expected to correct the sign of the determinant for M_2. There appears to be no dependable way to repair a faulty parameter set without checking determinants as the parameters are added (as in [2]), or previous knowledge of the correct values. Hence, it was impossible for us to create a valid Λ map for Ω_p, no matter how carefully we made our measurements. Figure 5 illustrates the changes made to the matrices, and the corresponding changes in the determinants.

Submatrix	Location of Parameter	Measured Value	Repaired Value	Difference	$\det(M_n)$
M_1	(2,12)	-1.3614e-4	-1.3704e-4	-9.0e-7	≈6.99e-17
M_2	(5,2)	-3.5483e-4	-3.3162e-4	2.321e-5	≈6.68e-12

Table 1. Comparison of repaired parameter values with measured ones.

A.4 Sensitivity of Parameters

In order to construct a valid Λ matrix from a parameter set, the set must be internally consistent. That is, the experimentally measured parameters must be close enough to the values in an actual Λ matrix. This can be checked (as shown in [2]) by examining the determinants of certain submatrices of the parameter set. We did an experiment to directly demonstrate how precisely the parameters must match in order to construct a viable Λ matrix from them.

Beginning with a parameter set with six digits of accuracy, we constructed a Λ matrix, and used it as the correct matrix to compare future ones to. We then removed on digit of precision, and reconstructed the Λ-map. We repeated this until there was only one digit left beyond the decimal point. As demonstrated by the graphs in Appendix C, this is when the Λ matrix became nearly unrecognizable.

It is clear from these results, that at least three signicant figures are necessary to create a valid parameter set. The parameters that we measured, du to experimental error, rarely had two significant figures of accuracy. Hence, it was impossible to reconstruct a valid Λ matrix.

B.1 Inverse Solver

An inverse solver is a computer program that computes the solution for the discrete inverse problem - see introduction. It takes a Λ matrix - the discrete approximation to the Λ_γ map - and computes the conductance of its corresponding network. The basic alogrithm for the computation is described by Curtis and Morrow in [1]. In this section, we describe the three versions of our inverse solvers, and discuss the techniques used in each version.

The first version of our inverse solver was an implementation of Curtis and Morrow's reconstruction scheme. This scheme reconstructed the network from a corner and went inward. The figure below illustrates stage 3 of the reconstruction of a 7 × 7 network.

Figure 6. Illustration for reconstruction of stage 3 of the 7 × 7 network.

In the figure above, the dashed line represents resistors whose conductances are to be computed at this stage. All the conductances to the right of the dashed one have been computed in the previous stages. Therefore, we can use them to compute the resistor that is dashed. We also note that at any particular stage, k, there are k level of resistors. Each level of resistors consists of one set of vertical and one set of hortizontal resistors. The reconstruction scheme for stage k can be divided into 5 steps.

1. Solving an system of linear equations.

   $$(I_k + \alpha_1 I_{4n} + \cdots + \alpha_k I_{4n-k+1})_j = 0$$

   where $j \in [1, 4n]$ and $j \neq \{1, 2, \ldots, k, 4n, 4n-1, \ldots, 4n-k+1\}$.

2. Compute the current entering the nodes at the top of the network by taking linear combinations of the columns of the Λ matrix.

3. Compute the voltage at the first level nodes right below the top nodes.

4. Compute the conductance of the resistor at the end of level 1.

5. Repeat the step 2 to 4 for the next level of nodes and resistors.

Following this scheme, after n stages, we have computed the conductance for half of the network. Then, we would rotate the network by 180 degree clockwise, and repeat the scheme. At this point, we have computed all the conductance over the network. The major disadvantage to this scheme is high inaccuracy in the last $n/2$ stages of calculation. We believe there are two reasons for the inaccuracy. First, since this method depends on all previous calculations, small errors at each stage accumulate and form big errors. Second, the range of the solution to the linear system becomes very large at stage n/2 or above. The magnitude of the solution starts around 1.00 volt and grows to 4^k volt at stage k calculation. For example, the range for the stage 7 calculation of a 10×10 network is [1, 16384]. With such a large range, one cannot accurately compute any current at the boundary nodes when taking linear combinations.

The second version of our inverse solver used four rotations instead of two. This eliminated the necessity of computing a majority of the resistors in the last $n/2$ inaccurate stages. This version of inverse solver turnd out to be fairly accurate. The experimental data showed that for a 5×5 network, the maximum error was about 10^{-13}. There are several drawbacks associate with this version of inverse solver. For example, this scheme didn't take advantage of the previous computed conductance at a nearby corner. This method also didn't allow the variation for the set of equations used to solve the linear system at Step 1. This last suggestion turned out to be a very big improvement in the reconstruction scheme - see next section for detail.

The third version of our inverse solver implemented both the variation of system of equations in Step 1 and the usage of previous computed conductance for later stage calculations. The variation of system included two parts:

1. Statregically moving the nodes where the solution to the linear system was for.
2. The change in system of equations used to solve the linear system.

This version was a very good inverse solver. From the experimental data, we believe this version of the inverse solver can work well on networks up to 15×15 with 7% maximum error. We have reproduced some experimental data in Appendix D.

B.2 Systems of Equations

A system of equations is a set of n equation sets. Each equation set is used to solve the linear system at the beginning of each stage. One of the biggest improvement to the accuracy in s recovering the conductance of a resistor network is being able to choose the right system of equations to solve the linear system in step 1 in the outline above. Experimental data showed that by choosing the right set of equation, the relative error of the recostruction reduced from $6.0 \cdot 10^{-6}$ to $7.8 \cdot 10^{-8}$. See appendix E for the summary of the worst errors for 4 different systems of equations. The questions arising from the experiment are: What make choosing the right set of equations so important in the reconstruction scheme, and how does one choose the right systems of equations? In this section, we have made an effort to answer these questions.

Recall the steps of reconstruction method above, the very first step at stage k calculation is to solve a $k \times k$ linear system. All the calculations, thereafter, depend on the solution to this linear system. Thus, the accuracy of the solution is very significant to the accuracy of the overall reconstruction. The main reason why choosing the right system of equations is so important is it makes the solutions to the linear system more accurate. The rows of this system are a subset of an overdetermined $2n \times 2n$ matrix. Each row corresponds to a boundary node at the left or bottom edge of the network.

Therefore, we have a lot of choices for the rows, but some combinations of rows produce singular systems (i.e. no solution.) For example, choosing the equations corresponding to two corner nodes will result in a singular $k \times k$ matrix. The remaining question is how to choose the right set of equations. Since there is no previous theory or conjecture about the best set of equations, we started out using a purely brute force method. We created two versions of equation testers for this purpose. Both testers could test systems for network that is 10×10 or smaller. One version generates all combinations at each stage using the nodes at the left side of the network. For example, at stage 3 of the reconstruction for a 5×5 network, the version one tester will test all combinations of 3 from the set of 5 nodes on the left side of the network. The other version does bascially the same thing as the first version, but instead of taking combinations of the equations on only one side of the network it takes equations associated to nodes at both the left and the bottom side of the network. In appendix F, we have reproduced the best systems of equations for different sizes of networks. For both versions of testers, the criteria to select the best set of equations at any stage is the Condition number, ZCond, returned from the DGECO routine in the Fortran 77 Linpack library. When DGECO is asked to solve $Ax = b$, it returns an approximation, z, to the solution, and a condition number, ZCond. This condition number, ZCond, tells how good z approximates x. Mathematically, the estimated condition number for the solution of a linear system is

$$Condition\ number = \frac{\|x - z\|}{\|x\|},$$

and it is the reciprocal of the Zcond returned by the Linpack routine. However, the above formula requires prior knowledge to the solution x. Thus, we must use some other method to approximate the condition number.

Let's suppose z is an approximation to $Ax = b$. Then, subtracting both sides of the equation by Az will result,

$$Ax - Az = b - Az$$

$$A \cdot (x - z) = b - Az$$

$$(x - z) = A^{-1} \cdot (b - Az)$$

At this point, we need to choose a norm for A that has the property: $\|Ax\| \leq \|A\|\|x\|$. (The Compatible Norm is one.) Applying the property just stated, we have

$$\|(x-z)\| = \|A^{-1} \cdot (b - Az)\| \leq \|A^{-1}\| \cdot \|(b - Az)\|$$

Now, divide the left side of the inequality by $\|Ax\|$, and the right side of the inequality by $\|b\|$.

$$\frac{\|(x-z)\|}{\|Ax\|} \leq \frac{\|A^{-1}\| \cdot \|(b-Az)\|}{\|b\|}$$

Since, $\|Ax\| \leq \|A\| \cdot \|x\|$, we can write $\|A\| \cdot \|x\|$ instead of $\|Ax\|$ in the above and still preserve the inequality.

$$\frac{\|(x-z)\|}{\|A\|\|x\|} \leq \frac{\|A^{-1}\| \cdot \|(b-Az)\|}{\|b\|}$$

Multiplying both side of the inequality by $\|A\|$, we have arrived at the upper bound of the Condition number.

$$\frac{\|(x-z)\|}{\|x\|} \leq \|A\| \cdot \|A^{-1}\| \cdot (\frac{\|(b-Az)\|}{\|b\|})$$

By repeating the above process with slight modification in steps, we will obtain another inequality for the lower bound of the Condition number:

$$\frac{1}{\|A\| \cdot \|A^{-1}\|} \cdot (\frac{\|b-Az\|}{\|b\|}) \leq \frac{\|(x-z)\|}{\|x\|}$$

Rearranging some terms, we finally get both the upper and lower bound for the condition number without requiring any prior knowledge of the real solution x.

$$\frac{1}{\|A\|\|A^{-1}\|} \cdot (\frac{\|Az-b\|}{\|b\|}) \leq \frac{\|x-z\|}{\|x\|} \leq \|A\|\|A^{-1}\| \cdot (\frac{\|Az-b\|}{\|b\|})$$

The DGECO routine in the Fortran Linpack library probably use something similar to compute the inverse of the condition number stated above.

After we tested different sized networks for the best system of equations, we discovered a pattern that allowed us to pick a system of equations that is close to the best system. The pattern is outlined as follows: Suppose we have an $n \times n$ network, and the boundary nodes numbered from 1 to $4n$, starting at the upper right corner. See figure 2 for an example. We always pick node $n+1$. Starting from the second stage to the (n/2 - 2) stage, we will pick node $3n-1$ along with node $n+1$. ¿From the third to the (n/2 - 2) stage, we pick the rest of the nodes by dividing the interval from $n+1$ to $3n-1$ into $k-1$ segments, where k is the stage number, and pick the node corresponding to the endpoints of the segments. From the (n/2 - 1) stage on, we pick node $3n$ instead of node$3n-1$, and also pick either node $2n-1$ or $2n$ along with node $n+1$. Beside the three nodes, we pick other nodes alternating from node $n+1$ forwards and node $3n$ backwards.

We have used this pattern to create the system of equation for a 10 × 10, a 15 × 15 and a 20 × 20 network. When comparing the results of the reconstruction for these networks, we have found our system is always better than the traditional system - which is form by taking successive nodes starting from the 1st one in the left edge.

V. Conclusion

After testing all the phases in reconstructing the physical network, we have found Curtis and Morrow's algorithm to reconstruct a resistor network is extremely hard to implement, given the resources at our disposal. We have proven the uniqueness property of the discrete forward problem. We have shown that the Λ_γ map is extremely insensitive to changes in the region of interest. In the reconstruction process, we learned new techniques to implement the inverse solver, as well as the program to reconstruct Λ matrices. We also found that reconstructing a Λ matrix from a given parameter set is extremely difficult - almost impossible. Although we failed to reconstruct the physical network from experiments, with more and better resources, we believe the reconstruction is possible.

Appendices

I. Appendix A

Table 2. Measured values vs. computed values for the potential at the interior nodes, when boundary node 1 is set to 6.29 volt and node 2 to 15 is grounded.

Node :	Measured value	Computed value
1	1.925	1.853189040
2	0.639	0.624769945
3	0.239	0.229750352
4	0.083	0.081268028
5	0.655	0.639041826
6	0.439	0.431407467
7	0.233	0.223877258
8	0.096	0.092798643
9	0.236	0.235501981
10	0.230	0.225770220
11	0.150	0.146354888
12	0.071	0.067450833
13	0.083	0.080941138
14	0.097	0.093722088
15	0.070	0.068544525
16	0.035	0.033693801

II. Appendix B

Table 3. Physical and computed measurements for the Λ matrix.

- The followings are the physical measurements (in mA):

-41.20	6.00	2.04	0.74	0.74	0.81	0.61	0.30	0.29	0.60	0.80	0.73	0.70	2.06	5.69	17.60
5.74	-39.2	6.37	2.17	2.16	2.03	1.32	0.62	0.60	1.15	1.32	0.86	0.82	1.99	3.85	5.80
2.09	6.44	-38.3	6.00	6.00	3.73	1.97	0.83	0.78	1.31	1.09	0.61	0.59	1.27	1.98	2.00
0.75	2.17	5.64	-41.3	17.8	5.90	2.05	0.73	0.66	0.82	0.59	0.31	0.30	0.61	0.84	0.72
0.75	2.17	5.63	17.9	-41.3	5.90	2.05	0.73	0.68	0.82	0.59	0.31	0.30	0.61	0.84	0.72
0.85	2.08	3.78	6.00	6.00	-38.2	6.70	2.10	1.97	2.00	1.27	0.63	0.60	1.14	1.34	0.82
0.62	1.34	1.98	2.20	2.07	6.70	-38.00	5.90	5.60	3.75	1.94	0.85	0.82	1.33	1.13	0.59
0.31	0.63	0.83	0.73	0.73	2.09	5.90	-40.40	17.00	5.90	2.03	0.75	0.72	0.84	0.61	0.30
0.30	0.61	0.80	0.71	0.70	2.02	5.37	17.30	-38.90	5.70	1.96	0.73	0.69	0.81	0.58	0.29
0.62	1.17	1.33	0.90	0.84	2.01	3.78	5.90	5.60	-37.70	6.50	2.16	2.07	2.03	1.29	0.60
0.82	1.34	1.10	0.60	0.60	1.28	1.95	2.10	1.95	6.50	-37.40	5.90	5.33	3.75	1.93	0.79
0.74	0.87	0.62	0.31	0.31	0.63	0.85	0.80	0.72	2.14	5.90	-40.70	17.30	6.10	2.08	0.72
0.71	0.83	0.59	0.30	0.30	0.60	0.81	0.72	0.69	2.05	5.60	17.30	-39.70	5.90	1.99	0.69
2.10	2.04	1.29	0.62	0.62	1.14	1.34	0.85	0.81	2.02	3.76	6.10	5.90	-38.30	6.60	2.02
5.80	3.90	2.00	0.85	0.85	1.35	1.13	0.61	0.58	1.29	1.93	2.08	2.10	6.60	-38.30	5.90
17.80	5.58	2.02	0.73	0.72	0.82	0.59	0.30	0.28	0.59	0.79	0.72	0.69	2.02	5.90	-40.30

- The followings are the computed values (in mA): • Note: the following values have been truncated.

-40.22	5.81	2.08	0.77	0.74	0.87	0.63	0.31	0.31	0.65	0.86	0.76	0.76	2.22	5.91	17.47
5.72	-37.8	6.44	2.25	2.15	2.13	1.36	0.63	0.64	1.22	1.39	0.88	0.88	2.14	4.03	5.95
2.06	6.47	-37.5	6.02	5.76	4.04	2.07	0.84	0.86	1.41	1.16	0.63	0.63	1.35	2.05	2.14
0.76	2.25	6.02	-41.1	17.4	6.18	2.22	0.76	0.78	0.91	0.65	0.33	0.33	0.66	0.90	0.79
0.74	2.18	5.81	17.6	-40.4	5.97	2.14	0.74	0.75	0.88	0.63	0.32	0.32	0.64	0.87	0.76
0.87	2.15	4.06	6.23	5.95	-38.8	6.68	2.16	2.21	2.15	1.37	0.65	0.65	1.22	1.42	0.90
0.63	1.37	2.08	2.23	2.13	6.68	-38.00	5.99	6.13	4.12	2.12	0.90	0.90	1.44	1.20	0.65
0.30	0.63	0.84	0.76	0.73	2.15	5.95	-40.4	17.6	5.87	2.13	0.76	0.76	0.87	0.62	0.32
0.31	0.64	0.86	0.78	0.75	2.19	6.08	17.6	-40.4	6.00	2.18	0.78	0.78	0.89	0.64	0.32
0.64	1.22	1.41	0.91	0.87	2.14	4.10	5.87	6.01	-38.8	6.75	2.28	2.28	2.18	1.39	0.67
0.85	1.39	1.15	0.64	0.61	1.35	2.09	2.11	2.16	6.70	-37.9	5.93	5.93	3.97	2.05	0.88
0.75	0.88	0.62	0.33	0.31	0.64	0.88	0.76	0.77	2.26	5.94	-40.3	17.7	5.92	2.15	0.78
0.76	0.88	0.63	0.33	0.31	0.64	0.89	0.76	0.78	2.27	5.95	17.7	-40.9	5.94	2.16	0.78
2.19	2.14	1.35	0.66	0.63	1.21	1.42	0.87	0.89	2.17	3.99	5.94	5.95	-38.4	6.66	2.27
5.76	3.98	2.02	0.89	0.85	1.39	1.17	0.61	0.63	1.37	2.04	2.13	2.13	6.58	-37.6	5.98
17.2	5.95	2.13	0.79	0.75	0.89	0.65	0.32	0.32	0.66	0.88	0.78	0.78	2.27	6.06	-40.5

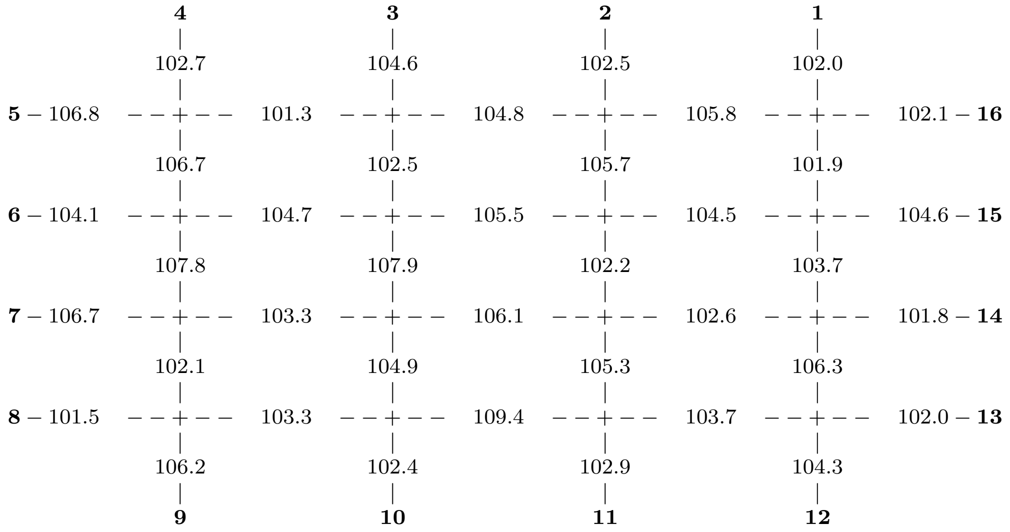
III. Appendix C

As explained in section II.B, the Λ matrix is relatively insensitive to changes in the function of conductances, γ. The following graphs are the results of an experiment performed on a 15×15 network. They are the differences between a "control" matrix, and the matrices created by slightly altered networks. A clear difference was first apparent when 64 of the 480 conductances were changed (graph 2a).

The following graphs are the reconstructed Λ matrices from parameter sets which have been truncated to limited precision.

IV. Appendix D

In this appendix, we have reproduced reconstruction results from version 1, 2, and 3 inverse solver. In the following data, the network beginning reconstruct is a 4 × 4 network.



```
Network reconstucted by version 1 inverse solver.
  ***** Conductance of the network *****
  Hort. conductor at   2  1   is   106.80000000000
  Hort. conductor at   2  3   is   101.30000000001
  Hort. conductor at   2  5   is   104.09424390244
  Hort. conductor at   2  7   is   105.80000000001
  Hort. conductor at   2  9   is   102.10000000000
  Hort. conductor at   4  1   is   102.10000000000
  Hort. conductor at   4  3   is   100.70605468749
  Hort. conductor at   4  5   is   94.064737650567
  Hort. conductor at   4  7   is   101.10228487335
  Hort. conductor at   4  9   is   104.60000000000
  Hort. conductor at   6  1   is   106.70000000000
  Hort. conductor at   6  3   is   95.429614679827
  Hort. conductor at   6  5   is   102.88222388658
  Hort. conductor at   6  7   is   100.35287804880
  Hort. conductor at   6  9   is   101.80000000000
  Hort. conductor at   8  1   is   101.50000000000
  Hort. conductor at   8  3   is   103.89999999999
  Hort. conductor at   8  5   is   109.82053045187
  Hort. conductor at   8  7   is   103.70000000000
  Hort. conductor at   8  9   is   102.00000000000
  Vert. conductor at   1  2   is   102.70000000000
  Vert. conductor at   1  4   is   104.60000000000
  Vert. conductor at   1  6   is   102.50000000000
  Vert. conductor at   1  8   is   102.00000000000
  Vert. conductor at   3  2   is   106.70000000000
  Vert. conductor at   3  4   is   98.205756097583
  Vert. conductor at   3  6   is   101.83355989181
  Vert. conductor at   3  8   is   101.89999999999
  Vert. conductor at   5  2   is   113.99394531249
  Vert. conductor at   5  4   is   95.332373865792
  Vert. conductor at   5  6   is   96.022027721780
  Vert. conductor at   5  8   is   106.94712195123
  Vert. conductor at   7  2   is   102.10000000001
  Vert. conductor at   7  4   is   103.32946954813
  Vert. conductor at   7  6   is   111.46645094641
  Vert. conductor at   7  8   is   106.30000000001
  Vert. conductor at   9  2   is   106.20000000000
  Vert. conductor at   9  4   is   102.40000000000
  Vert. conductor at   9  6   is   102.90000000000
  Vert. conductor at   9  8   is   104.30000000000

Network reconstucted by version 2 inverse solver.
  ***** Conductance of the network *****
  Hort. conductor at   2  1   is   106.80000000000
  Hort. conductor at   2  3   is   102.70000000000
  Hort. conductor at   2  5   is   104.80000000000
  Hort. conductor at   2  7   is   105.80000000000
  Hort. conductor at   2  9   is   102.10000000000
  Hort. conductor at   4  1   is   104.10000000000
  Hort. conductor at   4  3   is   104.69999999999
  Hort. conductor at   4  5   is   105.85448470023
  Hort. conductor at   4  7   is   105.65000000001
  Hort. conductor at   6  1   is   106.70000000000
  Hort. conductor at   6  3   is   103.19999999999
  Hort. conductor at   6  5   is   110.25876737313
  Hort. conductor at   6  7   is   102.60000000001
  Hort. conductor at   6  9   is   101.80000000000
  Hort. conductor at   8  1   is   101.50000000000
  Hort. conductor at   8  3   is   103.30000000000
  Hort. conductor at   8  5   is   109.40000000000
  Hort. conductor at   8  7   is   103.69999999999
  Hort. conductor at   8  9   is   102.00000000000
  Vert. conductor at   1  2   is   106.80000000000
  Vert. conductor at   1  4   is   104.10000000000
  Vert. conductor at   1  6   is   102.50000000000
  Vert. conductor at   1  8   is   102.00000000000
  Vert. conductor at   3  2   is   106.70000000000
  Vert. conductor at   3  4   is   104.69999999999
  Vert. conductor at   3  6   is   105.65000000001
  Vert. conductor at   3  8   is   105.80000000000
  Vert. conductor at   5  2   is   109.39999999999
  Vert. conductor at   5  4   is   97.888271481797
  Vert. conductor at   5  6   is   126.26906097309
  Vert. conductor at   5  8   is   104.80000000000
  Vert. conductor at   7  2   is   103.30000000000
  Vert. conductor at   7  4   is   104.89999999999
  Vert. conductor at   7  6   is   102.60000000001
  Vert. conductor at   7  8   is   106.29999999999
  Vert. conductor at   9  2   is   106.20000000000
  Vert. conductor at   9  4   is   102.40000000000
  Vert. conductor at   9  6   is   102.90000000000
  Vert. conductor at   9  8   is   104.30000000000


Network reconstucted by version 3 inverse solver.
  ***** Conductance of the network *****
  Hort. conductor at   2  1   is   106.80000000000
  Hort. conductor at   2  3   is   101.30000000000
  Hort. conductor at   2  5   is   104.80000000001
  Hort. conductor at   2  7   is   105.80000000000
  Hort. conductor at   2  9   is   102.10000000000
  Hort. conductor at   4  1   is   104.10000000000
  Hort. conductor at   4  3   is   104.69999999999
  Hort. conductor at   4  5   is   105.49999999999
  Hort. conductor at   4  7   is   104.45000000001
  Hort. conductor at   4  9   is   104.60000000000
  Hort. conductor at   6  1   is   106.70000000000
  Hort. conductor at   6  3   is   103.29999999998
  Hort. conductor at   6  5   is   106.10000000002
  Hort. conductor at   6  7   is   102.60000000002
  Hort. conductor at   6  9   is   101.80000000000
  Hort. conductor at   8  1   is   101.50000000000
  Hort. conductor at   8  3   is   103.30000000000
  Hort. conductor at   8  5   is   109.39999999999
  Hort. conductor at   8  7   is   103.70000000000
  Hort. conductor at   8  9   is   102.00000000000
  Vert. conductor at   1  2   is   102.70000000000
  Vert. conductor at   1  4   is   104.60000000000
  Vert. conductor at   1  6   is   102.50000000000
  Vert. conductor at   1  8   is   102.00000000000
  Vert. conductor at   3  2   is   106.70000000000
  Vert. conductor at   3  4   is   102.50000000001
  Vert. conductor at   3  6   is   105.65000000000
  Vert. conductor at   3  8   is   101.90000000000
  Vert. conductor at   5  2   is   107.80000000000
  Vert. conductor at   5  4   is   107.89999999999
  Vert. conductor at   5  6   is   102.20000000000
  Vert. conductor at   5  8   is   103.70000000000
  Vert. conductor at   7  2   is   102.10000000000
  Vert. conductor at   7  4   is   104.90000000000
  Vert. conductor at   7  6   is   105.30000000000
  Vert. conductor at   7  8   is   106.30000000000
  Vert. conductor at   9  2   is   106.20000000000
  Vert. conductor at   9  4   is   102.40000000000
  Vert. conductor at   9  6   is   102.90000000000
  Vert. conductor at   9  8   is   104.30000000000
```

V. Appendix E

Table . Summary of the maximum errors using 4 different systems of equations.

Location of Resistors	Tradition method of selecting	Choosing set of eqts. from 1 side	Choosing set of eqts. from 2 sides	Choosing set of eqts. from 2 sides.
10,11	$6.00 \cdot 10^{-6}$	$1.06 \cdot 10^{-7}$	$7.80 \cdot 10^{-8}$	$9.81 \cdot 10^{-8}$
12,11	$3.11 \cdot 10^{-7}$	$1.80 \cdot 10^{-7}$	$1.98 \cdot 10^{-8}$	$1.86 \cdot 10^{-8}$
11,10	$6.00 \cdot 10^{-6}$	$1.45 \cdot 10^{-7}$	$5.22 \cdot 10^{-8}$	$6.78 \cdot 10^{-8}$
11,12	$2.86 \cdot 10^{-7}$	$1.97 \cdot 10^{-7}$	$1.75 \cdot 10^{-8}$	$6.63 \cdot 10^{-9}$

VI. Appendix F

The following are the optimized systems for different size networks:

```
For 4 x 4 network:
  5   5   5   5
  0  12   6   6
  0   0   8  11
  0   0   0  12

Node 5 to  8 are on the left side of the network, and
Node 9 to 12 are on the bottom side of the network.

For 5 x 5 network:
  6   6   6   6   6
  0  15   7   7   7
  0   0  10  13   8
  0   0   0  15  13
  0   0   0   0  14

Node  6 to 10 are on the left side of the network, and
Node 11 to 15 are on the bottom side of the network.

For 10 x 10 network:
  11  11  11  11  11  11  11  11  11  11
  0   30  23  15  13  12  12  12  12  12
  0    0  30  26  24  16  14  13  13  13
  0    0   0  30  28  25  24  16  14  14
  0    0   0   0  30  29  27  25  25  16
  0    0   0   0   0  30  29  28  26  25
  0    0   0   0   0   0  30  29  27  27
  0    0   0   0   0   0   0  30  28  28
  0    0   0   0   0   0   0   0  29  29
  0    0   0   0   0   0   0   0   0  30

Node 11 to 20 are on the left side of the network, and
Node 21 to 30 are on the bottom side of the network.
```

References

[1] E. B. Curtis and J. A. Morrow, Determining the Resistors in a Network, SIAM, J. Appl. Math., 50 (1990), pp. 918-930.

[2] E. B. Curtis and J. A. Morrow, The Dirichlet to Neumann Map for a Resistor Network, SIAM, J. Appl. Math.